

# 画像からメインの5色を抽出するシステム

## 開発経緯

私がこのシステムを開発したのは本来時間をかけるべきではない作業を効率化することです。私はよく演劇公演の宣伝映像や舞台に投射する映像の作成を依頼されます。この時に手間となるのが、動画内で使う色の決定です。特に依頼数の多い宣伝映像やタイトル投射用の動画はチラシと与えるイメージを一緒にするために、チラシのメインカラーとなるべく同じ色を使用します。しかし、完成したチラシの画像しかいただけないことも多く、メインカラーを再現するのにとても手間取っておりました。

そこで、この作業にあてる時間を短縮できれば、デザインや構成に時間を割くことができ、動画のクオリティアップにつながると考え、開発をはじめました。

## このシステムでできること

1. 入力された画像のパスから画像を読み込む
2. 読み込んだ画像から主に使われている5色を抽出する
3. 先程の5色の反対色を割り出す
4. 抽出した5色を反対色の色番号とともにカラーパレットのように表示できる画像を生成して、別ウィンドウで表示する
5. キーボードのEを押すとそのままウィンドウが消える。Sを押すと、パソコンの画像フォルダに表示した画像を保存する

## 使用した言語・ライブラリー

【言語】

Python

【ライブラリー・モジュール】

OpenCV(主に画像の読み込み・書き出しに使用)

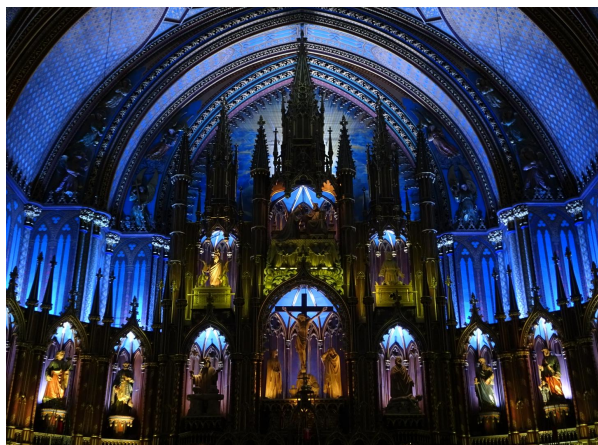
Pillow・PIL(主に色画像の生成やテキスト入力に使用)

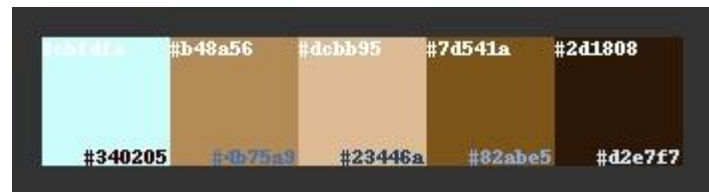
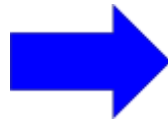
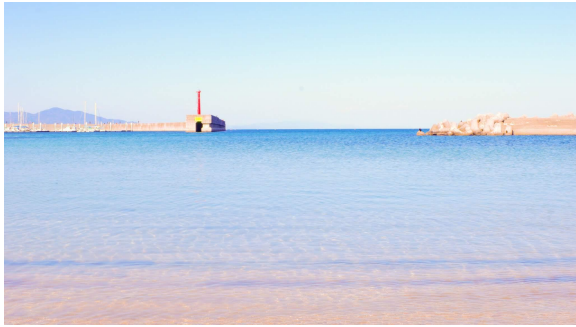
scikit-learn(機械学習ライブラリー、画像内で使用している色の分類に使用)

os(生成した画像の保存に使用)

NumPy(保存用のファイル名作成の乱数生成に使用)

## テスト結果





## 苦労したところ

- 画像生成を行うためにJupyter notebookを初めて使用したこと。
  - 使い方がわかるまでひたすら何度も使い方を調べました。
- PillowからOpenCVへの変換
  - 画像を別ウィンドウに表示させて、キーボード入力によって次の動作を決定するために画像データをPillowからOpenCVに変換する必要がありました。
  - その前の表示させる画像が出来上がった時点で正常に出力されることはわかっていたため、別ウィンドウに表示させる部分が問題であるのは確実でした。そのため、OpenCVとPillowで扱うことのできる画像データが違うかもしれないと仮説し、Googleでしらべはじめました。
  - 調べた結果、仮説は概ね正解でした。Pillowのまま画像を保存することもできましたが、どうしてもキーボードで保存するかどうか決定しなかったため、OpenCVへの変換を組み込みました。

## 今後の改良点

- Webアプリ化
  - Jupyter notebookを起動する手間を省きたいので、webアプリ化を目指したいです。
- 5色の抽出方法
  - 上記の海の写真と結果からもわかるように、差し色が反映されないため、反映されるような抽出方法を見つけたいです。